Technical Report

UCAM-CL-TR-833
ISSN 1476-2986

Number 833

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Practice-led design and evaluation of a live visual constraint language

Alan F. Blackwell, Ignatios Charalampidis

May 2013

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

Technical reports published by the University of Cambridge
Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Abstract

We report an experimental evaluation of Palimpsest, a novel purely-visual programming language. A working prototype of Palimpsest had been developed following a practice-led process, in order to assess whether tools for use in the visual arts can usefully be created by adopting development processes that emulate arts practice. This initial prototype was received more positively by users who have high self-efficacy in both visual arts and computer use. A number of potential usability improvements are identified, structured according to the Cognitive Dimensions of Notations framework.

1. Introduction

Palimpsest is a novel purely-visual programming language inspired by early data structure-oriented languages including FORTH and LISP, as well as interactive graphical programming systems such as Sketchpad (Sutherland 1963) and SmallTalk (Kay 1996). Palimpsest is intended for use on touch interface devices such as tablets, where emerging text-free and gesture-based interaction conventions provide new motivation to create purely-visual programming paradigms. The core abstraction in Palimpsest is a stack of layers, where each layer contains a graphical object or image (Figure 1). Programs are created by re-ordering and defining constraint relationships between these layers.

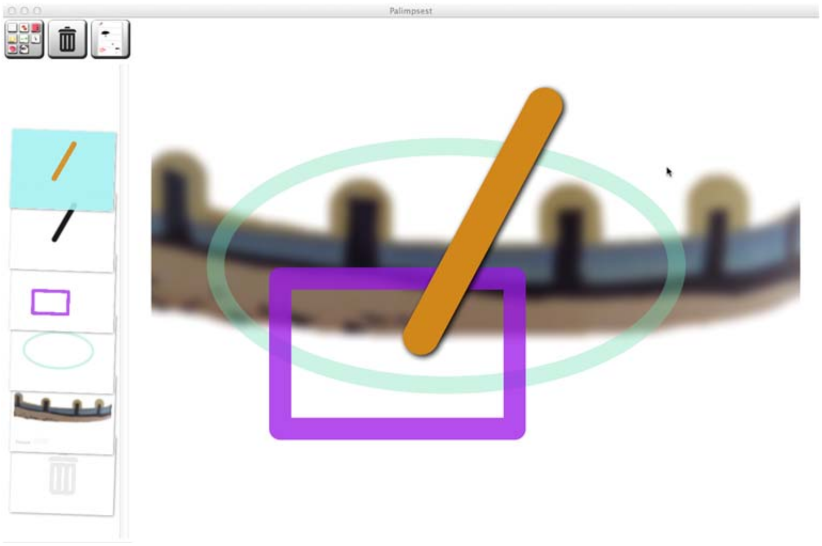


Figure 1 – the main Palimpsest window, with the stack of layers visible at the left. Small buttons at the top (from left to right) create a menu layer, delete the current layer, or collapse layers into a collection.

As a novel general-purpose programming paradigm, Palimpsest does not assume a specific application domain. Nevertheless, support for images as first-class objects, and text-free gesture interaction, suggests a possible audience of “visual thinkers”, or practitioners in the digital visual arts. The design was also informed by a series of collaborative projects developed through the Crucible network, creating software tools for use by a wide range of artists including musicians (Nash & Blackwell 2008), sculptors (Gernand et al 2011) and dancers (Church et al 2012). The development process therefore adopted an approach more typical of practice-led arts research rather than conventional software engineering (Rust, Mottram & Till 2007, Blackwell in press), emphasising the insight that can be achieved through use of “craft” tools (modern IDEs with extensive support for agile coding and refactoring) to develop a novel design through iterative prototyping rather than prior analysis (Woolford, Blackwell et al 2010).

2. Evaluation Strategy

Although seldom articulated so explicitly, it seems likely that many novel programming languages and environments are developed in this manner (e.g. see Seibel 2009). However, there is a tension between this common practice and the objectives of the end-user programming research community, which aims to apply the methods of user-centred design to software development tools (e.g. Ko et al 2011). User-centred design methods include evidence-based insight into user needs, empirical approaches to usability evaluation, and use of theoretically coherent analytic methods as a formative element of the design process (Gould & Lewis 1985). User-centred methods relevant to programming tools are developed and discussed in research communities such as the Psychology of Programming Interest Group (e.g. Hoc et al 1990).

In following a practice-led iterative approach to development, even if it is the case that this approach is tacitly adopted in other programming language research, there was a significant likelihood that Palimpsest might not meet the specific objectives of an end-user programming system. If viewed as an end-user programming language specifically *for* visual artists, rather than simply one inspired by their work practices, would this development process be an effective one?

One approach to testing that research question would be to compare the usability of Palimpsest to a language such as Processing, which was designed specifically for use by visual artists (Reas & Fry 2007). However, Processing is a relatively conventional language, supporting a complete textbook introduction to basic programming motivated by applications in the visual arts. Applications similar to the examples in the Processing text are introduced to Palimpsest users within the span of a comparatively short interactive tutorial. But in contrast to Processing, Palimpsest users do not acquire transferrable knowledge of text language syntax from this tutorial, and have no means to extend the computational facilities built-in to the system. From this perspective, Palimpsest appears less comparable to a programming language, but rather like a computational “sketchbook” that allows users to explore image transformations interactively.

Given that the intended usage of Palimpsest is so open-ended, controlled experimental tasks have relatively poor external validity as evaluations of its “sketchbook” functionality. The current Palimpsest prototype also presents problems with respect to the target host platform. As with systems such as Sketchpad and Smalltalk, Palimpsest was intentionally designed to exploit hardware resources beyond those that are commercially available at present, and current tablet devices do not provide sufficient memory and native graphics library support to execute a basic Palimpsest program. For this reason, the study described here is more exploratory in nature, and emulates the intended touch manipulation style of the system with a conventional laptop computer and mouse.

3. Evaluation Criteria

The development of Palimpsest was motivated by an overall research goal in the user experience of programming, relating to previous work in *attention investment* (Blackwell, Rode & Toye 2009) and self-efficacy (Beckwith et al 2006). The general principle arising from this prior research is that people without prior programming experience often over-estimate both the costs and the risks associated with abstract specification of behaviour. This over-estimation of costs and risks is associated with low *self-efficacy*, a psychometric measure of a person's confidence that they are able to act effectively in a particular domain (here, the domain of computer programming). The attention investment theory of abstraction use predicts that people with low self-efficacy in programming are more likely to prefer direct manipulation strategies to automated strategies. However, that preference results in habits of computer use that further reinforce low self-efficacy.

One strategy to address this problem is to support casual direct manipulation operations, that can be gradually replaced or augmented with abstract automations. Casual and exploratory actions are described in the end-user programming and educational literature as “tinkering” (Beckwith et al 2006), but they can also be compared to the activity of *sketching* in art and design practice (Eckert et al 2012). This connection between sketching and abstraction seems to offer a promising way of integrating insights from end-user programming and studies of digital creativity (Church, Nash &

Blackwell 2010). In particular, creative experiences of digital abstraction are associated with *liveness* (Tanimoto 1990), for example in the performance practice of live coding (Blackwell & Collins 2005).

The design goals of Palimpsest, therefore, are to address the obstacle of low self-efficacy in abstraction use, by encouraging sketch-like direct manipulation as live tinkering, leading to experimentation with more abstract specification of behaviour. The remainder of this report tests that objective by recruiting an experimental sample where there are broad variations in programming self-efficacy and familiarity with creative sketching. It will then be possible to assess how these variations correspond to differences in usage of Palimpsest. In order to directly compare these two attributes, we have taken an existing measure of computer self-efficacy (derived from Compeau & Higgins 1995), and compared this to a measure of self-efficacy related specifically to visual expression (Hickman & Lord 2010). Although not previously used in digital creativity research, this second measure has closely related objectives to that of the Compeau and Higgins questionnaire – it assesses the extent to which a person is confident in their own ability to create visual representations.

In previous research, we have noted that a wide variety of electronic and software products support programming-like abstractions (Blackwell, Hewson & Green 2003), including information-hiding, classification and repetition. The same is true of advanced and professional drawing products such as GIMP and Adobe Photoshop. In these products, the elements of an image can be organized into layers, including grouping and dependency relationships between the layers. Some potential applications of Palimpsest might extend the ways that layers are used in these applications.

Use of software abstractions is associated with notations that define the abstraction, and allow users to modify, extend, or utilize the abstractions. The usability of these notational systems can be analysed using the Cognitive Dimensions of Notations framework (Blackwell & Green 2003). That framework emphasizes the fact that different notational systems can be more or less well suited to particular kinds of user activity, and that suitability for a particular activity can be described as a particular profile of dimensions. When supporting creative users, and sketch-like activity, the profile associated with exploratory design is likely to be particularly relevant. It is therefore reasonable to ask whether the abstraction-supporting facilities of Palimpsest are well suited, in the view of users, to the kind of exploratory design usability profile that is expected of a sketching tool.

Research Questions

Drawing on the evaluation strategy and criteria presented above, we summarise our research questions as follows.

1. Our goal in this research is to assess how usable Palimpsest is, among a range of potential users that differ widely in the two forms of self-efficacy.

2. We also want to explore whether previous experience with abstraction in the context of computer drawing tools modifies their response to Palimpsest.

3. Finally, we wish to explore whether Palimpsest seems to support sketch-like usage, by collecting feedback from the sample user group elicited with the Cognitive Dimensions of Notations questionnaire (Blackwell & Green 2000).

5. Participants

We recruited 10 participants from among postgraduate students registered for degrees in different departments at the University of Cambridge. Five of these were recruited from professions in which a high degree of skill in programming would be expected, and five from professions in which a high degree of skill in visual arts would be expected (architecture, graphics, civil engineering). We expected that there would be a range of artistic ability among the “programmers”, and also that there would be a range of programming ability among the “artists”. The overall design of the sample was intended to maximize the range of variability, while retaining comparable levels of general professional experience/education. Table 1 gives a summary of the 10 participants, classified according to those recruited for expected ability in programming (*PPn*) or arts (*PAn*).

Participant	Background	Experience with drawing software
PA1	Architecture	AutoCAD, VectorWorks, Photoshop, Illustrator, Sketchup
PA2	Architecture	Photoshop, Paint, AutoCAD, Revit, 3DsMax,SolidWorks
PA3	Civil engineer	AutoCAD, Paint
PA4	Architecture	AutoCAD, 3DsMax,Illustrator,Photoshop, Form2,Rhino,Sketchup
PA5	Games graphics designer	Illustrator, Photoshop, Flash, Pixlr
PP1	Masters student in Computer Science	Art Rage, Photoshop, Paint
PP2	Masters student in Computer Science	Paint
PP3	Masters student in Computer Science	Paint
PP4	Software Engineer	Paint, Photoshop
PP5	Masters student in Computer Science	Photoshop, Paint, Flash, Illustrator

Table 1. Experimental participants, with background in visual arts (PA1-PA5) or programming (PP1-PP5) and indicating previous experience with drawing applications that might incorporate abstract features.

6. Experimental Procedure

All participants completed three questionnaires before starting to use Palimpsest:

a) A questionnaire assessing familiarity with drawing tools as reported in Table 1

b) Computing self-efficacy questionnaire (based on Compeau & Higgins 1995)

c) Visual arts self-efficacy questionnaire (based on Hickman & Lord 2010)

The three questionnaires are included as appendices of this report.

Participants then worked through the introductory tutorial that is incorporated in the current version of Palimpsest (task A). This tutorial is divided into two parts – an introduction to basic concepts and manipulation techniques, followed by descriptions of more advanced features. Although the early stages of the tutorial proceed slowly, and are designed with the expectation that all users will complete them, the later stages make it clear that users may prefer to exit the tutorial and return to it later, after gaining more experience with the system. Participants in the experiment were free to quit the tutorial any time they wished.

After leaving the tutorial, participants were given the opportunity freely to explore the facilities of Palimpsest, experimenting with their own ideas (task B). Initiating and evaluating creative play in a controlled experimental context suffers from some obvious threats to external validity, so we chose to follow a procedure that had successfully been used to assess a musical synthesis application, by asking musicians to improvise freely for a period of 15 minutes (Stead, Blackwell & Aaron 2012). The experimenter (the second author) also prompted participants during this time, suggesting other features that they may like to play with. Participants were given the option of continuing to play for as long as they wished, or to finish at any time.

After they finished using Palimpsest, participants completed a modified version of the Cognitive Dimensions of Notations questionnaire (Blackwell & Green 2000).

Typical durations of the experimental phases were 10 minutes to complete the initial questionnaires, 35-45 minutes on task A (following the Palimpsest tutorial), 15 minutes on task B (free play time), and 15 minutes for the post-test questionnaire.

The Palimpsest introductory tutorial introduces users to the features of the system in a phased manner. However, at the half-way point of the experiment, it became clear that this tutorial, although explaining individual features in a reasonably gradual way, did not provide participants with a good understanding of the potential applications of Palimpsest. The second author therefore modified the experimental procedure to include an initial introduction, in which he demonstrated some of the interesting visual effects that can be achieved with Palimpsest. Participants PA3, PA4, PA5, PP4 and PP5 received this additional training (note that this applied to different numbers in each group – this is discussed further in the conclusion of this report).

7. Comparison of computer and arts groups

Having recruited participants with a range of professional orientations, who were expected to have a range of self-efficacy in both visual arts and computing, we wished to make a comparison between those whose visual arts self-efficacy was higher than their computer self-efficacy and vice-versa. We therefore computed a metric that compared self-efficacy in visual arts (SEVA) to self-efficacy in computer use (SEC). The contrast between the two can be expressed as the difference (SEVA – SEC) Dividing the sample at the median value for this metric identified those participants having relatively high self-efficacy in visual arts rather than computing:

- Group A (PA1,PA2,PA4,PA5,PP5) had relatively high self-efficacy in visual arts compared to computing
- Group B (PA3, PP1, PP2, PP3, PP4) had relatively high self-efficacy in computing compared to visual arts

We then analysed the usability issues identified with the CDs questionnaire, in order to see which issues were considered more salient by the members of each of these groups. The overall findings are summarized in Table 2.

CD	Higher SE in Arts	Higher SE in Computing
Visibility	1	2
Viscosity	2	1
Provisionality	3	2
Abstraction Management	4	3
Error Proneness	4	3
Hidden Dependencies	4	3
Role - Expressiveness	3	2
Closeness of mapping	2	4

Table 2: Numbers of usability issues identified by different self-efficacy groups

Group A - higher self-efficacy in visual arts:

Abstraction Management: Most participants in this group found the abstract notations challenging. PA5 and PA2 thought that they could be helpful when creating more complex sketches, while PA4 felt that they could be useful when he became more familiar with them.

Hidden Dependencies: This was one of the most confusing parts of Palimpsest. Participants made several mistakes because they didn't know which layer was connected with what viewport. All participants suggested that the connections should be more obvious. They also mentioned that the connected layers do not follow a tree hierarchy in the same order as the stack overview, making connections harder to find.

Error Proneness: The most common error that participants encountered was dragging onto the wrong point. PA1 stated that she didn't know what the correct actions actually were, so was not sure whether she had made more mistakes or not. PA2 was irritated by mixing of visual elements between layers. PA5 constantly selected the wrong layer (e.g. the colour connected to a different layer than the one he wanted). PA4, PA5 and PP5 mentioned that they could not undo actions, although they also noted that deleting a colour connected to another layer retained that connection, even though the colour "did not exist" on the stack.

Group B - higher self-efficacy in computing:

Abstraction Management: As with participants in group A, participants in this group were confused by the abstract features of Palimpsest and how to use them. PP2, PP3 and PP4 suggested they should be accompanied with more information on what effect they have because it was impossible "to understand what they do by just looking at them".

Hidden Dependencies: Again, participants were confused about the connections between various parts of the systems. PA5 suggested that this problem would be solved if "when hovering your mouse over the connection-viewport, an indication of where it lies in the stack overview was given".

Closeness of mapping: PP1, PP3 and PP4 did not like the way that the menu was also presented as a layer, especially when manipulating that layer gave them the impression that they had done something wrong. They suggested that a conventional menu (e.g. in the form of a drop-down list) should be used instead.

Error Proneness: Three of the participants had difficulties with drag and drop behaviour as in group A. However, they did not mention that this is a feature that needs improvement but they suggested that having an undo button in the system would be more useful.

Summary

Despite being a tool intended to be usable by artists in particular, the "artists" in group A were just as likely to report usability problems. However, we noticed that when completing the tutorial, most participants in group B (higher self efficacy in programming) immediately started to experiment with system features, ignoring the instructions of the tutorial. As a result of this, they became lost more easily, and were obliged to re-read the tutorial. As found by Beckwith et al (2006), novice users with high self-efficacy were more likely to "tinker", and completed tasks more quickly, but this did not lead to better understanding.

8. Combined arts and computing self-efficacy

It became apparent in the course of the experiment that those participants most interested in Palimpsest were the ones with high self-efficacy in *both* visual arts and computer programming. Rather than making a comparison between the two groups, we therefore computed a further metric, measuring the *combined* self-efficacy with regard to visual arts and computing (SEVA + SEC).

We again separated our sample into two groups divided by the median value:

- Group A (PA2, PA4, PA5, PP1, PP2) have higher combined self efficacy in visual arts and computing.
- Group B (PA1, PA3, PP4, PP3, PP5) have lower combined self efficacy in arts and in computer programming.

As seen in Figure 2, those in group A tended to spend less time in the basic tutorial. They spent more time exploring abstract features and the advanced features of Palimpsest. They also spent more time in task B when given freedom to experiment with Palimpsest (differences in means not significant). One member of group A (PP1) chose to work through the whole of the advanced tutorial, reviewing all capabilities of Palimpsest rather than simply the introduction to basic facilities.

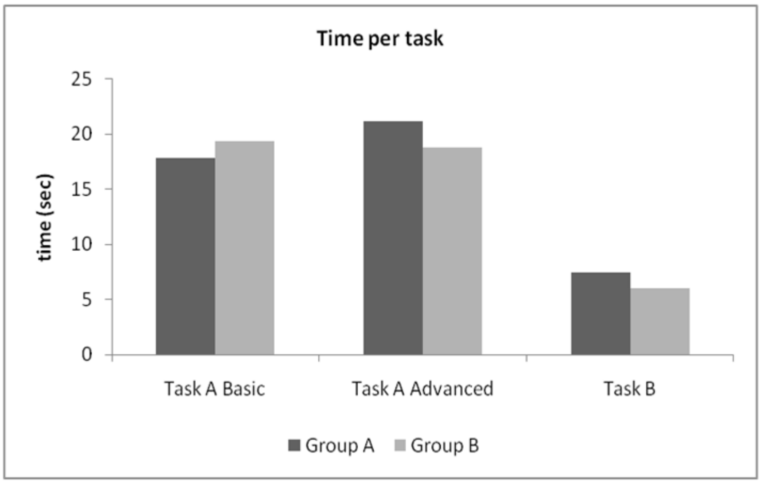


Figure 2. Mean number of minutes spent on each task by participants in Group A (high combined self-efficacy) and Group B (low combined self-efficacy)

Participants in group A were impressed by the animation capabilities of Palimpsest, and they considered the abstract notations to be useful when creating larger visualisations or entertaining animations. They encountered few difficulties in handling the layers, and on the contrary found them really useful. An exception was PP2, who we note had not previously used any drawing tool other than Paint. He found the concept of the layered drawing, and the associated abstractions, very difficult.

In contrast, participants in group B found Palimpsest difficult to use and had difficulties navigating the capabilities of the system. None of them considered Palimpsest to be a desirable tool for sketching. All except PP4 were confused by the way layers and viewports were connected and they often found themselves doing things that they did not understand. Most abandoned the advanced tutorial early, not because they thought they understand everything but because they were frustrated about their progress. As a result of this, they were not keen to spend much time in free play.

9. Impact of initial demonstration

As noted, Participants PA3, PA4, PA5, PP4 and PP5 received an additional training demonstration before starting to work through the Palimpsest tutorial. This consisted of an initial introduction by the experimenter, demonstrating some of the interesting visual effects that can be achieved with Palimpsest. Those who received additional training proceeded through the tutorials slightly faster than those who didn't, and continued with free play for slightly longer (figures 3 and 4 – differences not significant).

The additional training was introduced in part because users experienced difficulties selecting and moving the layers, and with the manipulation of the stack overview screen. After the introduction of additional training, there were no further complaints.

This reflects the novelty of the basic interaction paradigm in Palimpsest, which does not resemble other existing tools. For example, a common mistake made by participants before the introduction of the additional training was that they tried to drag and drop from one layer in the stack overview into another one in the stack overview. It was after a few attempts and guidance from the experimenter that they became familiar with the model of dragging a layer from the stack overview into the main area. Those who saw the additional training demonstration did not make this mistake.

Participants who received additional training particularly liked the demonstration of animation behaviour. This kind of behaviour is also incorporated by the first author in his public demonstrations of Palimpsest, in order to show its potential for creative applications. However, the conceptual basis of the operations required to make an animation are not introduced until later in the Palimpsest tutorial.

This tension between motivational applications and basic concepts is well known as a challenge in early programming education. Participants PA4 and PA5, having seen these animations during the additional training, said that they would like to use such a tool for creating animations, but did not see this as being related to “sketching” (i.e., they appreciated the abstract potential of the system, but did not see the benefit of integrating these functions with the techniques they might normally use for direct manipulation in an exploratory design context).

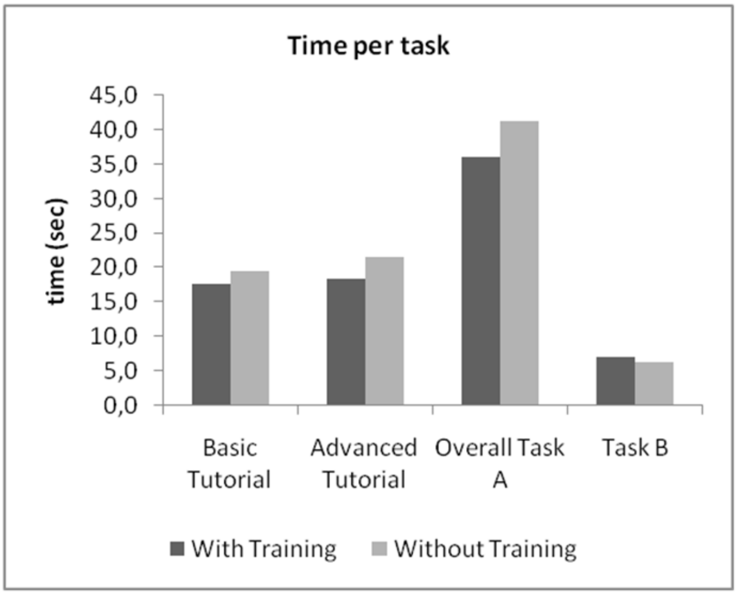


***Figure 3**: Time (minutes) spent in different phases of the experiment according to whether the user received a training demonstration or not.*

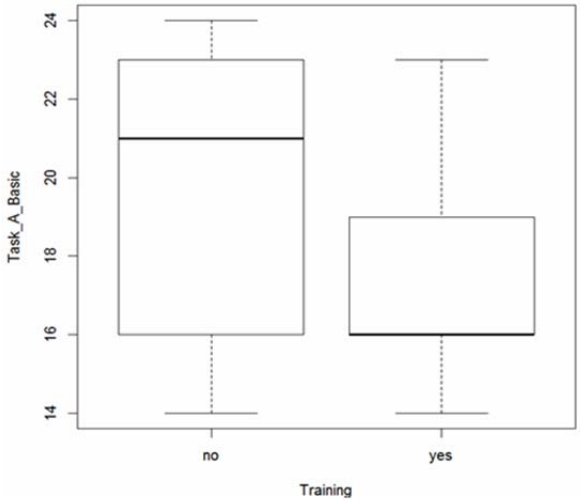


Figure 4: Box plot of time (minutes) spent in Task A comparing groups in which the users received a training demonstration or not

Another observation that should be taken into account for future work is that participants felt that the tutorial was too long, especially task A. Consequently, they were too tired to focus on a new task - which was the free experimentation time. Many participants were also frustrated with their relatively limited understanding of Palimpsest, and were reluctant to experiment.

10. Conclusions

The design of this experiment (conducted as a class exercise during a course on Usability of Programming Languages) has resulted in an unfortunate confound between two independent variables: first, the recruitment of participants from two different professional backgrounds; and second, the introduction of an additional training demonstration that was not equally balanced across the two groups. This second manipulation has resulted in a degree of variance within the experimental self-efficacy groups that has obscured the mean differences in behaviour between those groups, compounded by the fact that it was applied to different numbers in each group. Some of the specific behavioural changes observed appear to have been influenced by both the training demonstration, and variations in self-efficacy. We have reported these correlations, but are not able to attribute them definitively to one cause or the other.

Nevertheless, the participants' reaction to this first trial of a highly novel programming paradigm was encouraging. Apart from PP2, all participants found Palimpsest interesting and tried to use it in order to create something worthwhile. There appeared to be little difference in participants’ reaction to the system between groups having higher self-efficacy in arts in contrast to those having higher self-efficacy in programming. Users from both groups perceived similar usability issues. Participants from both groups were able to follow the tutorial and found Palimpsest amusing.

We have identified that the group with the most positive response to Palimpsest are those who have high self-efficacy *both* in visual arts and in computer use. Members of that group advanced more quickly through the basic tutorial and spent more time trying the advanced features of Palimpsest. They also spent more time experimenting with Palimpsest and tried to create something creative. Users in this group found the underlying abstractions more helpful, seemed to understand them better, and provided more useful feedback for the improvement of Palimpsest.

With regard to the exploration of practice-led design methods as a technique for development of novel programming tools, it is clear that this approach has resulted in a product that is best suited to users having a similar background and preferences to the developer himself (the first author). This is completely consistent with other fields of craft practice, where there is a traditional emphasis on the skilled craftsperson creating and maintaining his or her own tools as an essential element of craft competence. Indeed, there are traditions of craft training in which one of the accomplishments expected of an apprentice is the creation of a set of tools that will be used throughout his or her subsequent career.

This practice-led approach to the development of programming languages is typical of many research contexts, in which computer science researchers create languages that are used and appreciated by other computer science researchers. It is also characteristic of arts programming languages such as those used in live coding – for example Aaron’s *Overtone*, Sorensen’s *Extempore* or Magnusson’s *ixi lang*. Although typical of the genre, and apparently useful to these practitioners, it cannot be recommended as a substitute for user-centred design of end-user programming languages.

Just as live-coding languages appeal to music programmers and live-coding audiences, Palimpsest itself does have some appeal to users who have a combined interest in programming and visual arts. Future studies will investigate this group in more detail, as well as experimenting with ways in which its novel purely-visual programming paradigm might be adapted for use with a wider range of users.

11. References

Beckwith, L., Kissinger, C., Burnett, B., Wiedenbeck, S., Lawrance, J., Blackwell, A. and Cook, C. (2006). Tinkering and gender in end-user programmers' debugging. In *Proceedings of CHI 2006*, pp. 231-240.

Blackwell, A. and Collins, N. (2005). The programming language as a musical instrument. In *Proceedings of PPIG 2005*, pp. 120-130.

Blackwell, A.F. (in press). The craft of design conversation. To appear in M. Petre, A. van der Hoek and A. Baker (eds) *Software Designers in Action: A Human-Centric Look at Design Work.* Taylor & Francis.

Blackwell, A.F. & Green, T.R.G. (2000). A Cognitive Dimensions questionnaire optimised for users. In A.F. Blackwell & E. Bilotta (Eds.) *Proceedings of the Twelth Annual Meeting of the Psychology of Programming Interest Group (PPIG 2000)*, pp. 137-152.

Blackwell, A.F., Hewson, R.L. and Green, T.R.G. (2003) Product design to support user abstractions. In E. Hollnagel (Ed.) *Handbook of Cognitive Task Design*. Lawrence Erlbaum Associates. ISBN 0-8058-4003-6, pp. 525-545.

Blackwell, A.F., Rode, J.A. and Toye, E.F. (2009). How do we program the home? Gender, attention investment, and the psychology of programming at home. *International Journal of Human Computer Studies* 67, 324-341.

Church, L., Nash, C. and Blackwell, A.F. (2010). Liveness in notation use: From music to programming. In *Proceedings of the 22nd Annual Workshop of the Psychology of Programming Interest Group (PPIG 2010)*, pp. 2-11.

Church, L., Rothwell, N., Downie, M., deLahunta, S. and Blackwell, A.F. (2012). Sketching by programming in the Choreographic Language Agent. In *Proceedings of the Psychology of Programming Interest Group Annual Conference. (PPIG 2012)*, pp. 163-174.

Compeau, D.R. and Higgins, C.A. (1995). Computer self-efficacy: development of a measure and initial test. *MIS Quarterly* 19(2), 189-211.

Eckert, C., Blackwell, A.F., Stacey, M., Earl, C. and Church, L. (2012). Sketching across design domains: Roles and formalities. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 26(3), 245-266..

Gernand, B., Blackwell, A. and MacLeod, N. (2011). *Coded Chimera: Exploring relationships between sculptural form making and biological morphogenesis through computer modelling*. Crucible Network.

Gould, J. D., & Lewis, C. (1985). Designing for usability: Key principles and what designers think. *Communications of the ACM*, 28(3), 300–311.

Hickman, R. & Lord, S. (2010). An examination of adolescents' self-efficacy, engagement and achievement in representational drawing. *Australian Art Education* 32 (2),73-85

Hoc, J.-M. Green, T.R.G., Samurçay, R., and Gilmore, D.J. (Eds) (1990). *Psychology of Programming*. Academic Press.

Kay, A. (1996). The early history of Smalltalk. In *History of Programming Languages II*, T.J. Bergin, Jr. and R.G. Gibson, Jr., eds. ACM, New York. pp. 511-598.

Ko, A.J., Abraham, R., Beckwith, L., Blackwell, A.F., Burnett, M., Erwig, M., Lawrence, J., Lieberman, H., Myers, B., Rosson, M.-B., Rothermel, G., Scaffidi, C., Shaw, M., and Wiedenbeck, S. (2011). The State of the Art in End-User Software Engineering. *ACM Computing Surveys* 43(3), Article 21.

Nash, C. & Blackwell, A.F. (2008). Realtime representation and gestural control of musical polytempi. In A. Camurri, S. Serafin and G. Volpe (Eds), *Proc. 8th Int Conf on New Interfaces for Musical Expression (NIME'08).* Genova Italy June 4-8, pp. 28-33.

Reas, C. and Fry, B. (2007). *Processing: A programming handbook for visual designers and artists.* MIT Press.

Rust, C. Mottram, J. Till, J. (2007) *Review of practice-led research in art, design & architecture*. Arts and Humanities Research Council, Bristol, UK

Seibel, P. (2009). *Coders at work: Reflections on the craft of programming*. New York: Apress.

Stead, A.G., Blackwell, A.F. and Aaron, S. (2012). Graphic Score Grammars for End-Users. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME),* pp. 176-179.

Sutherland, I.E. (1963/2003). *Sketchpad, a man-machine graphical communication system*. PhD Thesis at Massachusetts Institute of Technology, online version and editors' introduction by A.F. Blackwell & K. Rodden. Technical Report 574. Cambridge University Computer Laboratory

Tanimoto S.L. (1990). VIVA: A Visual Language for Image Processing. *Journal of Visual Languages and Computing* 1(2), 127-139.

Woolford, K., Blackwell, A.F., Norman, S.J. & Chevalier, C. (2010). Crafting a critical technical practice. *Leonardo* 43(2), 202-203.

Appendix A. Pre-test questionnaire part 1 –experience with drawing software

Q1. What is your background? ....

Q2. Have you used any drawing software, before? YES NO

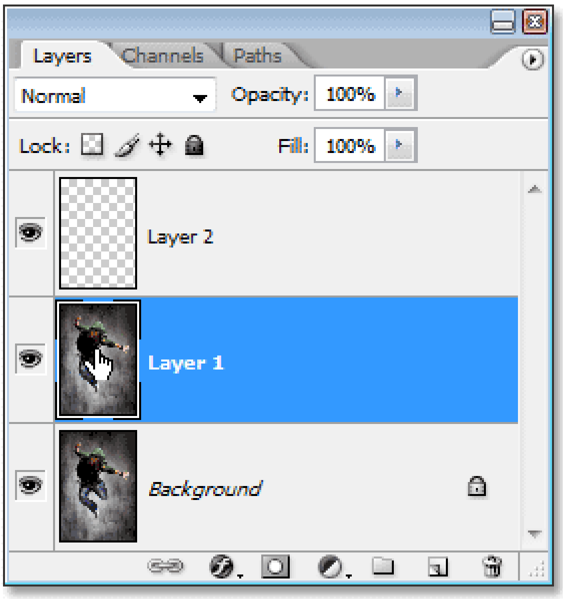
If yes, which ones? ....

....

Q3. Does your previous tool use layers? YES NO DON'T KNOW

If yes, do you use layers in your design? YES NO DON'TKNOW

If you are not familiar with what is a layer, does the following picture look familiar to you? Have you used something similar? YES NO



Normal

Opacity: 100%

Fill: 100%

If you asked NO in the previous question, you can skip questions Q4 to Q6.

Q4. In a grade of 1 to 5 (with 1 being not familiar at all, and 5 being expert at using them), how will you judge your efficiency in using layers?

1 2 3 4 5

Q5. What do you find difficult about layers? Describe in your own words.

Q6. Why do use layers in your sketches - design?

Q7. Have you used any visual-programming software tools? YES NO

THANK YOU FOR AGREEING TO TAKE PART IN THIS STUDY.

THERE ARE NO RIGHT OR WRONG ANSWERS. WE ARE ONLY INTERESTED IN YOUR HONEST OPINION.

Appendix B. Pre-test questionnaire part 2 – computer self-efficacy

Often in our jobs we are told about software packages that are available to make work easier. For the following questions, imagine that you were given a new software package for some aspect of your work. It doesn't matter specifically what this software package does, only that it is intended to make your job easier and that you have never used it before. The following questions ask you to indicate whether you could use this unfamiliar software package under a variety of conditions. For each of the conditions, please indicate whether you think you would be able to complete the job using the software package. Then, for each condition that you answered “yes,” please rate your confidence about your first judgment, by circling a number from 1 to 10, where 1 indicates “Not at all confident,” 3 indicates “Moderately confident,” and 5 indicates “Totally confident”.

FOR EXAMPLE, LOOKING AT QUESTION NUMBER 1
If you strongly feel confident that you are capable of completing a job on a new software tool with no one to give you instructions then place a circle round 5; if you feel moderately confident circle 3, if you are not feeling at all confident circle 1; circle 2 or 4 if you find yourself between the previous situations.

I COULD COMPLETE THE JOB USING A SOFTWARE PACKAGE....

Q1. if there was no one around to tell me what to do as I go.

Totally confident		Moderately Confident		Not at all
5	4	3	2	1

Q2. if I had never used a similar tool like it before.

Totally confident		Moderately Confident		Not at all
5	4	3	2	1

Q3. if I only had the software manuals for reference.

Totally confident		Moderately Confident		Not at all
5	4	3	2	1

Q4. If I had seen someone else using it before trying it myself.

Totally confident		Moderately Confident		Not at all
5	4	3	2	1

Q5. if I could call someone for help if I got stuck.

Totally confident		Moderately Confident		Not at all
5	4	3	2	1

Q6. if someone else had helped me get started.

Totally confident		Moderately Confident		Not at all
5	4	3	2	1

Q7. if I had a lot of time to complete the job for which the software was provided.

Totally confident		Moderately Confident		Not at all
5	4	3	2	1

Q8.if I had just the built-in facility for assistance.

Totally confident		Moderately Confident		Not at all
5	4	3	2	1

Q9. if someone showed me how to do it first.

Totally confident		Moderately Confident		Not at all
5	4	3	2	1

Q10. If I had used similar packages before this one to do the same job.

Totally confident		Moderately Confident		Not at all
5	4	3	2	1

THANK YOU FOR TAKING PART IN OUR QUESTIONNAIRE
Ignatios Charalampidis

Appendix B. Pre-test questionnaire part 3 – artistic self-efficacy

THANK YOU FOR AGREEING TO TAKE PART IN THIS STUDY. WE ARE INTERESTED IN YOUR THOUGHTS ABOUT ART.

THERE ARE NO RIGHT OR WRONG ANSWERS. WE ARE ONLY INTERESTED IN YOUR HONEST OPINION.

READ EACH OF THE STATEMENTS BELOW WHICH RELATE TO ART AND LEARNING - FOR EACH STATEMENT PLEASE INDICATE YOUR LEVEL OF AGREEMENT WITH EACH STATEMENT BY RINGING THE NUMBER NEAREST THE SCORE THAT BEST INDICATES HOW YOU FEEL MOST OF THE TIME.

FOR EXAMPLE, LOOKING AT STATEMENT NUMBER 1. If you strongly agree that you are capable of learning how to draw anything in art then place a circle round 5; if you agree circle 4; if you neither agree or disagree circle 3, if your disagree circle 2 and if you disagree strongly circle 1.

1. I am capable of learning how to draw anything in art if I make enough effort

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1

2. I am capable of learning how to draw something difficult in art

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1

3. I am capable of learning how to draw people in art

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1

4. I am capable of learning how to draw anything placed in front of me

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1

5. I am capable of learning how to draw buildings and landscapes

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1

6. I will ask for help from teachers when I am struggling in art lessons

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1

7. I will ask other students to help me when I get stuck in art

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1

8. I will ask a friend to help me when I get stuck in art lessons

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1

9. I am capable of finishing my art homework on time

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1

10. I am capable of working hard in art when there are other interesting things to do

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1

11. I am always able to concentrate in art lessons

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1

12. I am capable of understanding what is being asked of me

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1

13. I am capable of using the library and the internet to get information on artists

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1

14. I am capable of remembering information and instructions that the teacher gives me

Strongly agree Agree Neither agree/disagree Disagree Strongly disagree
5 4 3 2 1